**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Основы программирования»

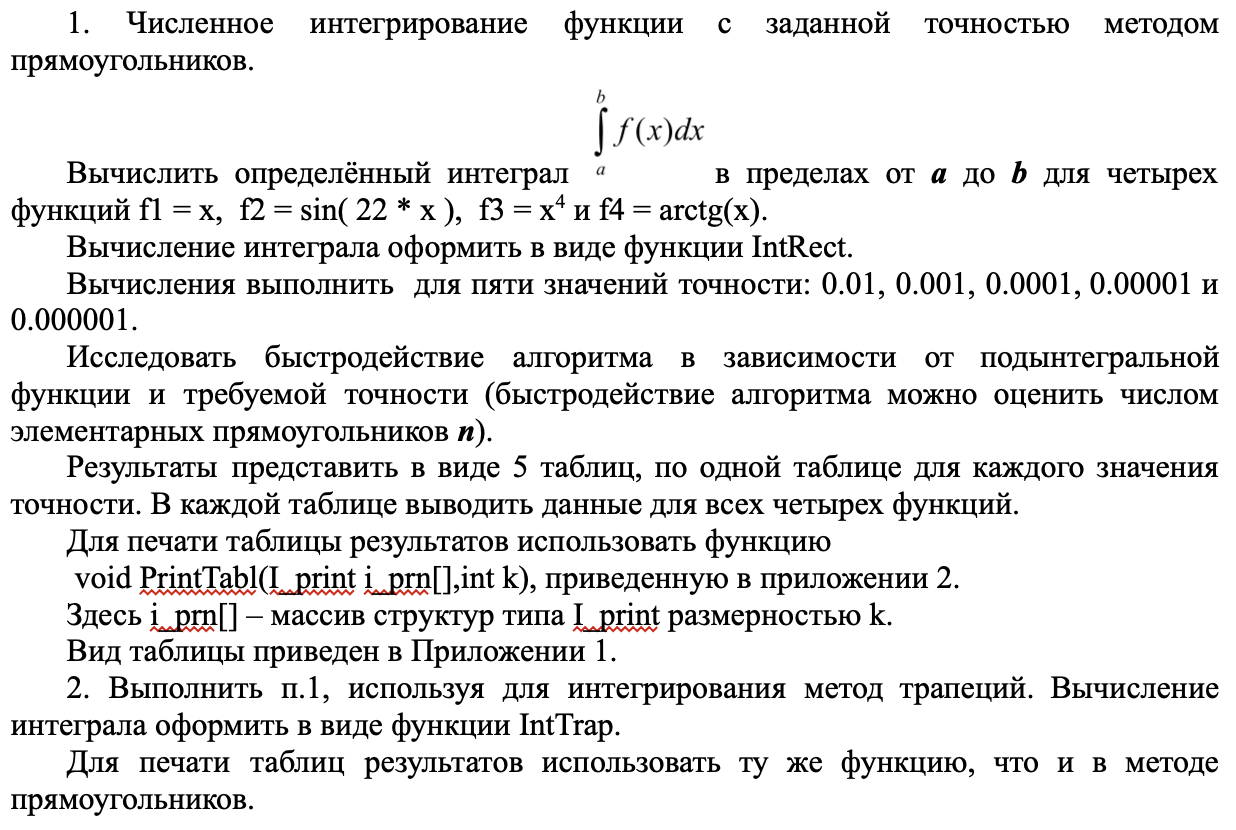
Отчет по лабораторной работе № 6

«Численное интегрирование функции»

| Выполнил: Чернев Николай Андреевич |  | Проверил: |
| --- | --- | --- |
| студент группы ИУ5-14Б |  | преподаватель каф. ИУ5 |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

Задание



Разработка алгоритма

Описание переменных

Структура I\_Print: (для вывода результатов)

* char \*name - математическая формула
* double i\_sum - полученное в функциях значение интеграла
* double i\_toch - точное значение интеграла
* int n - кол-во итераций

Функция вывода таблицы:

* const int m - кол-во столбцов таблицы
* int wh[m] - ширина столбцов таблицы
* char \*size[m] - заголовки столбцов
* int size[m] - длина строк в массиве size
* int i, j - счетчики циклов

Функция поиска интегралов методом прямоугольников:

* double dx - текущая ширина прямоугольников (по оси х)
* double s\_prev - предыдущая сумма площадей
* double s - текущая сумма площадей
* double r - текущая погрешность
* int &n - кол-во разбиений на прямоугольники

Функция поиска интегралов методом трапеций:

* double dx - текущая высота трапеций (по оси х)
* double x - левая граница текущей трапеции
* double s\_prev - предыдущая сумма площадей
* double s - текущая сумма площадей
* double r - текущая погрешность
* int &n - кол-во разбиений на трапеции

typedef double (\*TPF) (double); - переименование типа указателя на подынтегральную функцию

main:

* double a, b - левая и правая границы интегрирования
* I\_Print res[4] - массив результатов вычислений для разных функций
* TPF funcs[4] - массив указателей на функции
* char \*names[4] - массив названий математических функций
* double epses[5] - массив погрешностей
* double real\_values[4] - массив реальных значений интегралов
* int i, j - счетчики циклов

Описание функций

double f1(double x)-принимает х и возвращает значение ф-ции 1 в т.х

double f2(double x)-принимает х и возвращает значение ф-ции 2 в т.х

double f3(double x)-принимает х и возвращает значение ф-ции 3 в т.х

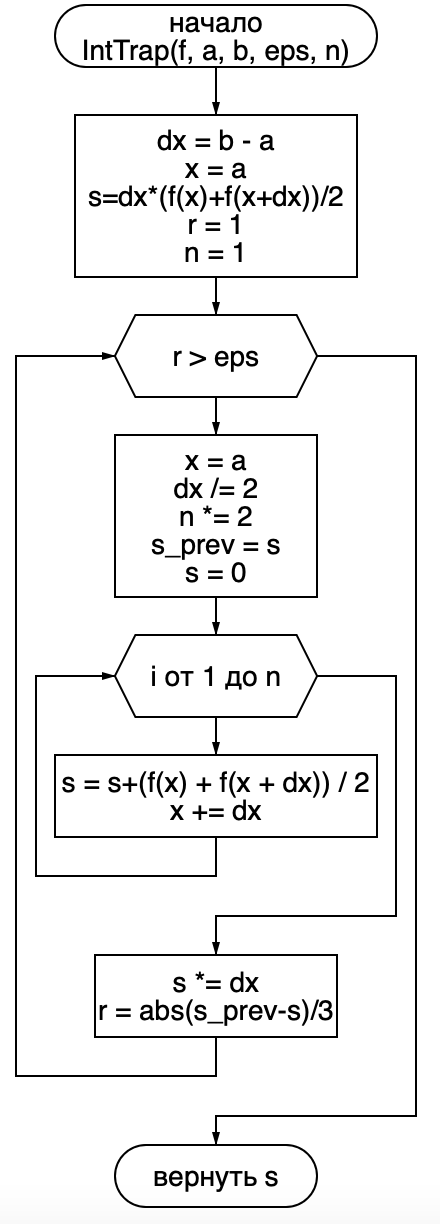
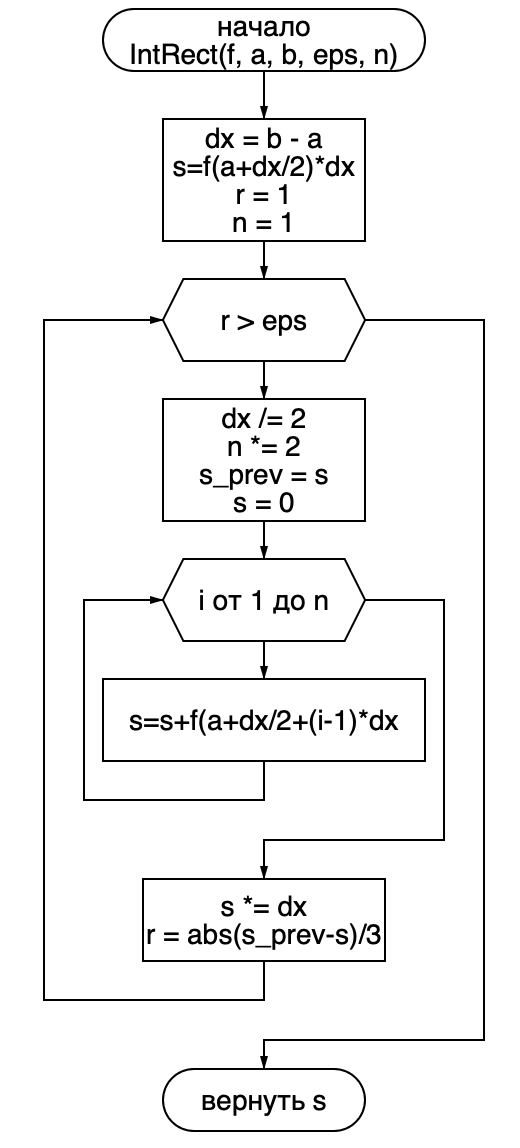
double f4(double x)-принимает х и возвращает значение ф-ции 4 в т.х

void PrintTabl(I\_Print i\_prn[],int k) - принимает k - кол-во строк в таблице и массив результатов вычислений интегралов и выводит на экран этот массив в виде таблицы, ничего не возвращает

double IntRect(TPF f, double a, double b, double eps, int& n) - принимает f - указатель на подынтегральную функцию, a и b - левую и правую границы интегрирования, погрешность eps и копию n - кол-ва разбиений; вычисляет интеграл методом прямоугольников и возвращает его

double IntTrap(TPF f, double a, double b, double eps, int& n) - принимает f - указатель на подынтегральную функцию, a и b - левую и правую границы интегрирования, погрешность eps и копию n - кол-ва разбиений; вычисляет интеграл методом трапеций и возвращает его

Блок-схема



Код программы

funcs.cpp

#include <cmath>

#include <iostream>

#include <iomanip>

using namespace std;

typedef double (\*TPF) (double);

double f1(double x){

return x;

}

double f2(double x){

return sin(22 \* x);

}

double f3(double x){

return pow(x, 4);

}

double f4(double x){

return atan(x);

}

struct I\_Print{

char\* name;

double i\_sum;

double i\_toch;

int n;

};

void PrintTabl(I\_Print i\_prn[],int k){

const int m=4;*//число столбцов таблицы*

int wn[m]={12,18,18,10};*//ширина столбцов таблицы*

char \*title[m]={"Function","Integral","IntSum","N "};

int size[m];

for(int i=0;i<m;i++)

size[i]=strlen(title[i]);

*//шапка таблицы*

cout<<char(218)<<setfill(char(196));

for(int j=0;j<m-1;j++)

cout<<setw(wn[j])<<char(194);

cout<<setw(wn[m-1])<<char(191)<<endl;

cout<<char(179);

for(int j=0;j<m;j++)

cout<<setw((wn[j]-size[j])/2)<<setfill(' ')<<' '<<title[j]

<<setw((wn[j]-size[j])/2)<<char(179);

cout<<endl;

for(int i=0;i<k;i++)

{*//заполнение таблицы*

cout<<char(195)<<fixed;

for(int j=0;j<m-1;j++)

cout<<setfill(char(196))<<setw(wn[j])<<char(197);

cout<<setw(wn[m-1])<<char(180)<<setfill(' ')<<endl;

cout<<char(179)<<setw((wn[0]-strlen(i\_prn[i].name))/2)<<' '<<i\_prn[i].name

<<setw((wn[0]-strlen(i\_prn[i].name))/2)<<char(179);

cout<<setw(wn[1]-1)<<setprecision(10)<<i\_prn[i].i\_toch<<char(179)

<<setw(wn[2]-1)<<i\_prn[i].i\_sum<<setprecision(6)<<char(179)

<<setw(wn[3]-1)<<i\_prn[i].n<<char(179)<<endl;

}

*//низ таблицы*

cout<<char(192)<<setfill(char(196));

for(int j=0;j<m-1;j++)

cout<<setw(wn[j])<<char(193);

cout<<setw(wn[m-1])<<char(217)<<setfill(' ')<<endl;

}

double IntRect(TPF f, double a, double b, double eps, int& n){

double dx = b - a;

double s\_prev, s = f(a + dx / 2) \* dx;

double r = 1;

n = 1;

while (r > eps){

dx /= 2;

n \*= 2;

s\_prev = s;

s = 0;

for (int i = 1; i <= n; i++){

s = s + f(a + dx / 2 + (i - 1) \* dx);

}

s \*= dx;

r = abs(s\_prev - s) / 3;

}

return s;

}

double IntTrap(TPF f, double a, double b, double eps, int& n){

double dx = b - a;

double x = a;

double s\_prev, s = dx \* (f(x) + f(x + dx)) / 2;

double r = 1;

n = 1;

while (r > eps){

x = a;

dx /= 2;

n \*= 2;

s\_prev = s;

s = 0;

for (int i = 1; i <= n; i++){

s = s + (f(x) + f(x + dx)) / 2;

x += dx;

}

s \*= dx;

r = abs(s\_prev - s) / 3;

}

return s;

}

header.h

#pragma once

#include <iomanip>

#include <iostream>

#include <cmath>

using namespace std;

typedef double (\*TPF) (double);

struct I\_Print{

char\* name;

double i\_sum;

double i\_toch;

int n;

};

void PrintTabl(I\_Print i\_prn[],int k);

double f1(double x);

double f2(double x);

double f3(double x);

double f4(double x);

double IntRect(TPF f, double a, double b, double eps, int& n);

double IntTrap(TPF f, double a, double b, double eps, int& n);

main.cpp

#include "header.h"

using namespace std;

typedef double (\*TPF) (double);

int main() {

double a = -1, b = 3;

I\_Print res[4];

TPF funcs[4] = {f1, f2, f3, f4};

char \*names[4] = {"y=x", "y=sin(22x)", "y=x^4", "arctg(x)"};

double epses[5] = {0.01, 0.001, 0.0001, 0.00001, 0.000001};

double real\_values[4] = {(b\*b - a\*a) / 2, (cos(22\*a) - cos(22\*b)) / 22, (pow(b, 5) - pow(a, 5)) / 5,

b \* atan(b) - a \* atan(a) - (log(b\*b + 1) - log(a\*a + 1)) / 2};

cout << "INT RECT\n";

for (int j = 0; j < 5; j++){

for (int i = 0; i < 4; i++){

res[i].n = 0;

res[i].name = names[i];

res[i].i\_toch = real\_values[i];

res[i].i\_sum = IntRect(funcs[i], a, b, epses[j], res[i].n);

}

cout << "Погрешность eps = " << epses[j] << '\n';

PrintTabl(res, 4);

cout << "\n\n";

}

cout << "INT\_TRAP\n";

for (int j = 0; j < 5; j++){

for (int i = 0; i < 4; i++){

res[i].n = 0;

res[i].name = names[i];

res[i].i\_toch = real\_values[i];

res[i].i\_sum = IntTrap(funcs[i], a, b, epses[j], res[i].n);

}

cout << "Погрешность eps = " << epses[j] << '\n';

PrintTabl(res, 4);

cout << "\n\n";

}

return 0;

}

Анализ результатов